

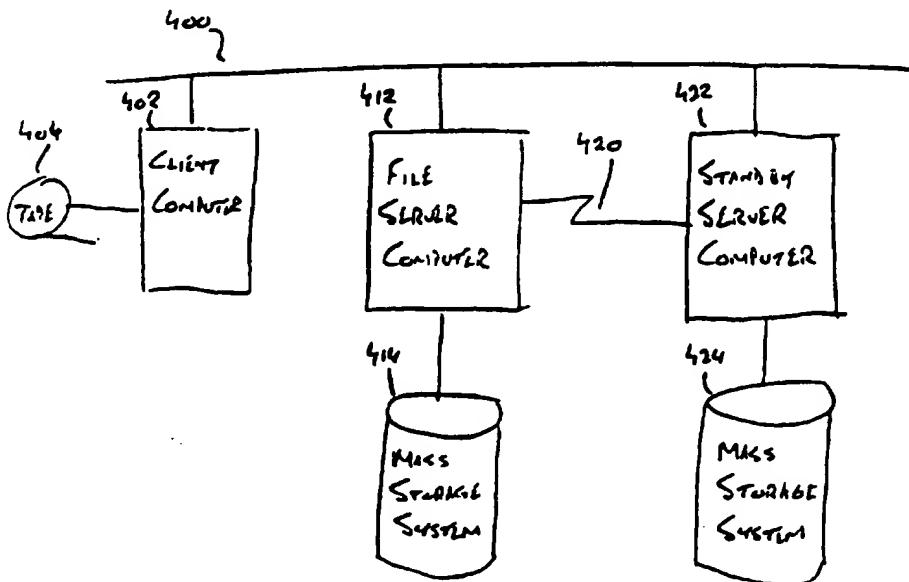
PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6: G06F 12/16, 12/08	A1	(11) International Publication Number: WO 96/12232 (43) International Publication Date: 25 April 1996 (25.04.96)
<p>(21) International Application Number: PCT/US95/13324</p> <p>(22) International Filing Date: 10 October 1995 (10.10.95)</p> <p>(30) Priority Data: 08/322,697 13 October 1994 (13.10.94) US</p> <p>(71) Applicant (for all designated States except US): VINCA CORPORATION [US/US]; 4000 Central Park East, 1815 South State Street, Orem, UT 84058 (US).</p> <p>(72) Inventors; and (75) Inventors/Applicants (for US only): OHRAN, Richard, S. [US/US]; 71 West 4750 North, Provo, UT 84604 (US). OHRAN, Michael, R. [US/US]; 109 South 200 East, Orem, UT 84058 (US).</p> <p>(74) Agents: CHRISTIANSEN, Jon, C. et al.; Van Cott, Bagley, Cornwall & McCarthy, Suite 1600, 50 South Main Street, P.O. Box 45340, Salt Lake City, UT 84145-0340 (US).</p>		<p>(81) Designated States: AL, AM, AT, AU, BB, BG, BR, BY, CA, CH, CN, CZ, DE, DK, EE, ES, FI, GB, GE, HU, JP, KE, KG, KP, KR, KZ, LK, LR, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SI, SK, TJ, TT, UA, US, UZ, VN, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG), ARIPO patent (KE, MW, SD, SZ, UG).</p> <p>Published With international search report. Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</p>

(54) Title: SNAPSHOT OF DATA STORED ON A MASS STORAGE SYSTEM



(57) Abstract

A method for providing a static snapshot, or image, of data stored on a mass storage system (104). At the start of the method, a preservation memory (106) is cleared and a virtual device is created. Whenever a write is to be performed on the mass storage system (104), a check is made of the preservation memory (106) to determine if it contains a block associated with the mass storage write address. If there is not, a copy of the block in the mass storage system (104) at the block write address is placed in the preservation memory (106). Whenever a read is to be performed on the virtual device, a check is made of the preservation memory (106) to determine if it contains a block associated with the virtual device read address. If there is such a block, that block is returned as the result of the virtual device read. Otherwise, the block at the virtual device block read address is returned as the result.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	GB	United Kingdom	MR	Mauritania
AU	Australia	GE	Georgia	MW	Malawi
BB	Barbados	GN	Guinea	NE	Niger
BE	Belgium	GR	Greece	NL	Netherlands
BF	Burkina Faso	HU	Hungary	NO	Norway
BG	Bulgaria	IE	Ireland	NZ	New Zealand
BJ	Benin	IT	Italy	PL	Poland
BR	Brazil	JP	Japan	PT	Portugal
BY	Belarus	KE	Kenya	RO	Romania
CA	Canada	KG	Kyrgyzstan	RU	Russian Federation
CF	Central African Republic	KP	Democratic People's Republic of Korea	SD	Sudan
CG	Congo	KR	Republic of Korea	SE	Sweden
CH	Switzerland	KZ	Kazakhstan	SI	Slovenia
CI	Côte d'Ivoire	LI	Liechtenstein	SK	Slovakia
CM	Cameroon	LK	Sri Lanka	SN	Senegal
CN	China	LU	Luxembourg	TD	Chad
CS	Czechoslovakia	LV	Latvia	TG	Togo
CZ	Czech Republic	MC	Monaco	TJ	Tajikistan
DE	Germany	MD	Republic of Moldova	TT	Trinidad and Tobago
DK	Denmark	MG	Madagascar	UA	Ukraine
ES	Spain	ML	Mali	US	United States of America
FI	Finland	MN	Mongolia	UZ	Uzbekistan
FR	France			VN	Viet Nam
GA	Gabon				

A STATIC SNAPSHOT OF DATA ON MASS STORAGE

Background of the Invention

Field of the Invention. This invention relates mass storage systems for digital computers, and in particular to a method for providing a static snapshot or image of a mass storage system.

Description of Related Art. It is desirable during the operation of a computer system with a mass storage system, such as a magnetic disk, to periodically make a backup copy of the data stored on the mass storage system to allow for recovery in the event of a failure of the mass storage system. This is commonly done by reading the data stored on the mass storage system and writing it to a magnetic tape.

However, if the data stored on the mass storage system is being updated by other programs as the backup copy is being made, the image of the data on the mass storage system written to tape may be inconsistent. This is because normal backup techniques either copy the blocks from the mass storage system sequentially to the linear-access tape, or walk the file system stored on the mass storage system, starting with the first block of the first file in the first directory and proceeding in order to the last block of the last file of the last directory. The backup program is not aware of updates performed to a block of the mass storage system after that block has been written to tape.

This problem of inconsistent data being written to tape is particularly likely to occur if the mass storage system is being used by a database management system, where an update may involve changing information stored on different parts of the mass storage system. If a database update is made while the backup tape is being written, the image of the database management system written to tape

will have the old values for any data already written to tape at the time of the database update, and the new values for any data written to tape following the database update. A restoration based on the tape image of the database would yield an inconsistent database.

Horton et al., United States Patent No. 5,089,958, which is hereby incorporated by reference in its entirety for the material disclosed therein, discloses a technique for producing an image of a mass storage system at any point in time after the technique is started. This is done by establishing a base image of the mass storage system at the start of the technique and a log indicating each change made to the mass storage system. An image at any point in time can then be produced by starting with the base image and making all the changes indicated in the log up to that point in time. To improve performance, the Horton system also provides for differential images so that the compilation of changes to form an image does not have to start with the base image.

There are two difficulties with using the technique of Horton to provide an image for backup operations. First, the technique is not designed to provide a static snapshot or image of the mass storage system, but to allow an image from any point in time to be created at some later time. This increases the complexity of the technique and requires the compilation of changes whenever a virtual image is desired.

The second difficulty with using the technique of Horton is that the log must store a copy of each change made to the mass storage system in order to produce an image of the mass storage system as it was at a specified time. This means that the size of the log can grow without bound, eventually exhausting the space available for its storage. At this point, updates to the mass storage system

are not possible without compromising the ability to produce an image from any previous point in time.

With many database systems or file systems, certain key blocks (such as master directory blocks) are frequently updated, perhaps with every update to any other block. A copy of these blocks must be written to the log each time they are changed. This will, of course, result in a very large log file, with many of the entries being copies of the key blocks as they changed over time.

Another approach to creating a static image of a mass storage system is possible if the mass storage system has the ability to produce a mirror, or identical copy, of one disk's data on a second disk. At the time the static image is needed, mirroring of data is stopped and the mirror disk is used as the static image. When the static image is no longer necessary (for example, when the tape backup has been completed), the two disks are resynchronized, by copying any changes made during the time mirroring was not active to the mirror disk, and mirroring is resumed.

This approach also has problems. Unless there are three or more disks mirroring the information on the main disk, when mirroring is stopped to produce the static image there is no longer the redundancy of mirrored disk or disks and updates can be lost if there is a disk failure. Furthermore, it requires an entire disk to be devoted to the storage of the static image.

But the major disadvantage of this mirror disk approach is the time necessary to restart mirroring after the static image is no longer needed. This requires updating the mirror disk with all the changes that have been made since mirroring was stopped. If a log of these changes is not available, this means that all the data on the mirror disk must be copied from the disk which has been

updated. For large disks such as would be found on a database system, this could take many hours.

For more general background reference materials, the reader is directed to United States Patent Nos.

3,444,528, 3,533,082, 4,141,066, 4,156,901, 4,164,017,
4,191,996, 4,351,023, 4,378,588, 4,403,303, 4,453,215,
4,459,658, 4,479,214, 4,483,001, 4,484,275, 4,486,826,
4,498,145, 4,507,751, 4,521,847, 4,581,701, 4,607,365,
4,608,688, 4,639,856, 4,648,031, 4,652,940, 4,654,819,
4,654,857, 4,674,038, 4,703,421, 4,703,481, 4,713,811,
4,727,516, 4,736,339, 4,750,177, 4,754,397, 4,878,167,
5,307,481, 5,235,700, 5,079,740, 4,823,256, 5,295,258,
5,157,663, 4,471,429, 4,530,052, 4,615,001, 4,941,087,
4,959,768, 4,979,108, 4,800,488, 4,656,596, 4,866,707,
3,771,137, 4,402,046, 4,423,414, 4,430,693, 4,477,882,
4,480,304, 4,562,535, 4,604,690, 4,630,224, 4,644,470,
4,648,061, 3,754,211, 3,824,547, 4,439,859, 4,445,214,
4,691,314, 3,754,211, 4,332,027, 4,516,121, 4,583,089,
4,646,300, 5,060,185, 5,123,099, 3,602,900, 3,665,173,
3,681,578, 3,735,356, 3,760,364, 3,761,884, 3,810,119,
3,828,321, 3,054,560, 3,303,474, 3,544,477, 3,623,014,
3,636,331, 3,725,861, 3,803,568, 4,012,717, 4,076,961,
4,099,241, 4,118,772, 4,358,823, 4,359,718, 4,455,645,
4,477,895, 4,589,090, 4,590,554, 4,610,013, 4,623,883,
4,639,852, 4,654,846, 4,680,581, 3,557,315, 3,636,331,
3,810,121, 3,812,469, 3,820,085, 3,828,321, 3,864,670,
3,865,999, 3,889,237, 3,959,638, 3,991,407, 4,073,005,
4,099,241, 4,152,764, 4,208,715, 4,228,503, 4,257,009,
4,270,167, 4,282,572, 4,318,173, 4,358,823, 4,371,754,
4,403,286, 4,428,044, 4,455,601, 4,530,051, 4,590,554, and
4,628,508 each of which is hereby incorporated by reference in their entirety for the material disclosed therein. The reader's attention is also directed to the following publications: Lyon, "Tandem's Remote Data Facility," IEEE

(1990); and Molina et al., "Issues in Disaster Recovery," IEEE (1990), each of which is incorporated by reference in its entirety for the material disclosed therein.

Summary of the Invention

It is an object of this invention to provide a static image of data stored on a mass storage system as it existed at a particular point in time.

This is accomplished by creating a virtual device that will appear as a mass storage device containing the static image. Write operations to the mass storage system are also intercepted by the method. Copies of blocks on the mass storage system are placed in a preservation memory whenever they are going to be changed by a write operation, unless an entry for that block is already in the preservation memory. During a read of the virtual device, the preservation memory is first checked, either directly or using a table of contents of the preservation memory, to see if it contains a copy of the block from the specified location. If the preservation memory has such a copy, that copy is returned as the result of the read. Otherwise, the block is read from the mass storage system.

It is a further object of the invention to reduce the amount of storage required to provide the static image. The technique of Horton requires the storage of all changes from the time the technique is started. The mirror disk technique requires storage equal to the size of the mass storage being imaged. In contrast, the method of the invention only requires storage equal to the number of mass storage blocks that have been changed since the static image was created.

It is a further object of the invention to reduce the time necessary for generating the static image and for returning to normal operation when the static image is no longer needed. Unlike the technique of Horton, where the

static image at a particular time needs to be compiled from the base image and log entries, all that is necessary for creating a static image using the method of this invention is to create the virtual device and establish the interception of writes to the mass storage system. No copying of data or compilation of an image is necessary.

When the static image is no longer necessary, the virtual device is removed from the system and the contents of the preservation memory deleted if recovery of that space is desirable. No synchronization to the mass storage system is necessary nor is it necessary to merge updates into a base image to create a new base image.

These and other features of the invention will be more readily understood upon consideration of the attached drawings and of the following detailed description of those drawings and the presently preferred embodiments of the invention.

Brief Description of the Drawings

Figure 1 is a block diagram of a representative digital computer configuration on which the preferred embodiment of the invention operates.

Figure 2 is a flow diagram showing the preferred steps of the method of the invention.

Figure 3 is a block diagram of a client-server configuration using the preferred method.

Figure 4 is a block diagram of the currently-preferred embodiment of the method in a client-server configuration.

Figure 5 is variant of the configuration of Figure 4.

Detailed Description of the Invention

Referring to Figure 1, which illustrates a representative computer configuration on which the method of the invention runs, it can be seen that digital computer 102 has connected to it mass storage system 104 and preservation memory 106. In some embodiments of the invention, association memory 108 may also be connected to digital computer 102.

Mass storage system 104 can be any writable block-addressable storage system, such as one or more disks or a partition of a disk. (If mass storage system 104 were not writable, such as a CD-ROM, its contents would not change and there would be no need for the invention of this application.) A partition of a disk can be a fixed area of a disk. The disks can store their information using magnetic, optical, or any other technique that allows writing and reading of data without departing from the scope and spirit of this invention.

In the currently-preferred embodiment of the invention, digital computer 102 is a PC-compatible computer based on an Intel X86 series or compatible processor and mass storage device 104 is a SCSI or IDE magnetic disk connected to digital computer 102 through an appropriate controller.

Preservation memory 106 can be an area in the random-access memory (RAM) of digital computer 102, one or more disks, a partition of a disk, or a file stored on a disk. Optimal selection of the implementation of preservation memory 106 depends of the number of blocks of the mass storage system that will be changed during the time the static image is needed. The use of RAM provides faster performance, but may limit the number of entries in the preservation memory. If the preservation memory runs out of space when a new entry must be stored, the method fails and the static image is no longer available or

remains in the state it was at the time the preservation memory ran out of space. It is important to note that if this occurs, no data from the mass storage system is lost, and the method can be restarted to produce a new static image.

Referring to Figure 2, which is a flow diagram showing the steps of the method, the method starts at step 202 when a static image of the mass storage system is desired. This can be indicated by the running of a special program, an operating system call, or an operator command, as appropriate for the operating system and application.

In step 202, preservation memory 106 is cleared. In general, this will consist of setting the control information describing the contents of preservation memory 106 to indicate that there are no valid entries in preservation memory 106.

In step 204, a virtual device appearing as a mass storage device is created. The method for creating a virtual device will depend on the particular operating system running on digital computer 102, but will be known by one skilled in the art of that particular operating system. In addition, it may be necessary in step 204 to configure the operating system so that the method of this invention intercepts any read or write operation directed to mass storage system 104. Again, how this is done will be dependent on the particular operating system running on digital computer 102.

In step 206, the method waits until there is a write operation directed to mass storage system 104 or a read operation directed to the virtual device created in step 204. In variants of the method, step 206 also reacts to a read operation directed to mass storage system 104 or a write operation directed to the virtual device created in step 204.

If the operation is a write to mass storage system 104, step 210 is entered. Using the mass storage write address specified in the write operation, step 210 determines if there is a block of data associated with that mass storage write address in preservation memory 106. If there isn't, step 212 of the method is executed. Otherwise, step 212 is skipped and step 214 is executed.

There are a number of ways for determining whether there is a block of data associated with the mass storage address in the preservation memory 106. In the currently-preferred embodiment of the invention, there is a block association memory 108 also connected to digital computer 102. (Block association memory 108 may be a separate memory connected to digital computer 102, or may be a portion of the RAM of digital computer 102.) Block association memory 108 is used to associate blocks stored in preservation memory 106 with the unique addresses of blocks on mass storage system 104. Block association memory 108 does this by containing entries indicating a unique address and the location in preservation memory 106 for the block associated with that unique address.

Entries in block association memory 108 can be stored unordered, in which case they must be linearly searched for a matching unique address. If no entry is found with a matching address, there is not a block in preservation memory 106 associated with that address. Alternatively, the entries could be stored ordered by unique addresses, in which case a binary search could be used to locate the matching entry in block association memory 108. A hashing scheme could also be used to find a matching entry.

The block association memory 108 can also be organized as an array with an element for each unique address of mass storage system 104. Each element of the

array stores a preservation memory location, or a special value that indicates that there is not a block in preservation memory 106 associated with that unique address.

The selection of a technique for storing entries in block association memory 108 depends on the characteristics of accessing the entries. Using an array provides the highest speed for accessing an entry or adding an entry corresponding to a block just copied into preservation memory 106, at the expense of a large block association memory 108. Ordering the entries by unique address provides faster access than for unordered entries, but requires more time when an entry is added to block association memory 108. In the currently-preferred embodiment, entries are stored unordered in block association memory 108.

In this discussion, the term block refers to the data stored at a particular location in mass storage system 104 or preservation memory 106. Blocks are generally of a fixed size (e.g. 512 bytes for disks used with MS-DOS), although blocks of different sizes, or variable sizes, are within the scope of this invention. On mass storage system 104, each block has a unique address, specified in read or write operations. A block in preservation memory 106 is a copy of a block of data stored in mass storage system 104, and that block in preservation memory 106 is associated with the unique address of the block in mass storage system 104 of which it is a copy.

For efficiency, it may be convenient to treat one or more contiguous blocks on mass storage system 104 as if it were a single, large block. Often operating systems perform their mass storage operations on contiguous blocks (called clusters in MS-DOS). The extensions to handle

clusters of blocks should be clear to one with ordinary skills in computer programming.

If block association memory 108 is being used, step 202 (clearing preservation memory 106) consists of removing all entries from block association memory 108 or setting them to the special entry that indicates that there is no block in preservation memory 106 associated with each unique address.

Returning to Figure 2, step 212 is executed if there is not a block associated with the mass storage write address in preservation memory 106. Step 212 places a copy of the block of data currently located at the mass storage write address in preservation memory 106, updating block association memory 108 as necessary. It is important to note that step 212 will be executed at most once for each unique address on mass storage system 104, since the next time step 210 tests to see if there is a block in preservation memory 106 associated with that mass storage write address it will find the copy made by step 212. Because of this, preservation memory 106 will contain only copies of blocks as they were when the method was started.

In step 214, the data to be written by the mass storage write operation is written to the location on mass storage system 104 specified by the mass storage write address. This completes the steps for a mass storage write, and step 206 is reentered to wait for the next operation.

If the operation is a virtual device read, step 220 is entered. Again, a check is made to determine if a block associated with the virtual device read address is in preservation memory 106. If there is such a block, step 224 is executed. If not, step 222 is executed.

Step 222 returns the data from the block in mass storage system 104 specified by the virtual device read

address as the result of the read operation. Step 224 returns the block from preservation memory 106 associated with the virtual address read address as the result of the read operation. This completes the steps for a virtual device read, and step 206 is reentered to wait for the next operation.

If the operation is a mass storage read, step 230 is entered, which returns the data from the block of mass storage system 104 specified by the mass storage read address as the result of the read operation. This completes the steps for a mass storage read, and step 206 is reentered to wait for the next operation.

It may be desirable to allow write operations to the virtual device, changing the image as specified by the write operations. For example, it may be necessary to write a different label or other control information on the virtual device image so the operating system can differentiate it from mass storage system 104.

If the operation is a virtual device write, step 240 is entered. Step 240 checks to see if the virtual device is read-only, and if it is step 242 is entered to return an appropriate error indication to the operating system or user.

Step 244 checks to determine if a block associated with the virtual device write address is in preservation memory 106. If there is such a block, step 248 is executed. If not, step 246 is executed. In step 246, the data from the virtual device write operation is placed in preservation memory 106, associated with the virtual device write address from the virtual device write operation. Block association memory 108 is updated as necessary. In step 248, the data from the virtual device write operation replaces the block associated with the virtual device write address of the virtual device write

operation. This completes the steps of the virtual device write, and step 206 is reentered to wait for the next operation.

While the description above describes the basic operation of the method of the invention, there are a number of other embodiments possible. For example, the same preservation memory 106 can be shared so that a second virtual device provides a snapshot image of a second mass storage system. In another embodiment, a second preservation memory and second virtual device can be used to provide a second image whose snapshot was taken at a different time of mass storage system 104.

The computer system running the method of the invention can also be used as a file server for client computers connected to it by a network or other means. As a file server, it can export its mass storage system, the virtual device created by the method, or both. Such a system is illustrated in Figure 3.

File server computer 312, with mass storage system 314, runs the method of the invention. It exports the virtual device (and probably mass storage system 314) to client computer 302, communicating over network 300. Computer 302 can run a tape backup program that copies the information from the exported virtual device to tape drive 304. No change is necessary for the tape backup program running on client computer 302, which sees the virtual device as just another mass storage device.

Figure 4 illustrates the currently-preferred configuration for running the method of the invention. Network 400 connects client computer 402, with tape drive 404, to file server computer 412, with mass storage system 414. File server computer communicates with standby server computer 422 over data link 420. Standby server computer 422 has mass storage system 424. Through software running

on file server computer 412 and standby server computer 422, as described in United States Patent application serial number 08/094,744, filed on July 20, 1993 and entitled "METHOD FOR RAPID RECOVERY FROM A NETWORK FILE SERVER FAILURE" (which is hereby incorporated by reference in its entirety), mass storage system 424 appears as a disk to file server computer 412 and mirrors the data on mass storage system 414. In the event of a failure of either file server computer 412 or mass storage system 414, standby computer 422 can be restarted as the file server.

In the configuration of Figure 4, standby server 422 runs the method of the invention, and can export the virtual device either to file server computer 412, which can then export it to client computers on network 400, or standby server 422 can directly export the virtual device to client computers. The virtual device can also be accessed by programs running on standby server 422.

Figure 5 illustrates a variant of the configuration of Figure 4. Instead of client computer 502 having a tape drive, as was the case for client computer 402, backup computer 532 has tape drive 534. Backup computer 532 communicates with standby server computer 522 over data link 530. Standby server computer exports mass storage system 524 to file server computer 512 (whether mirrored or not). Standby server computer exports the virtual device with the snapshot image of mass storage system 524 to backup computer 532.

Backup computer 532 can now copy the snapshot image of mass storage system 524 by reading the virtual device exported to it by standby server computer 522. Neither file server computer 512 nor standby server computer 522 has the overhead of the tape backup process, which can result in a degradation of performance if data compression needs to be performed before the data is

written to tape. Also, a fault in the tape backup program will not affect either file server computer 512 or standby server computer 522.

It is to be understood that the above described embodiments are merely illustrative of numerous and varied other embodiments which may constitute applications of the principles of the invention. Such other embodiments may be readily devised by those skilled in the art without departing from the spirit or scope of this invention and it is our intent they be deemed within the scope of our invention.

Claims

We claim:

1. A method for providing a static snapshot of data stored on a mass storage system, operating on a computer configuration including:

a digital computer executing the steps of the method;

a mass storage system connected to said digital computer, said mass storage system storing blocks of data having unique addresses; and a preservation memory connected to said digital computer, said preservation memory storing blocks of data associated with said unique addresses;

the method comprising:

(A) clearing said preservation memory so that no copies of blocks of data are in said preservation memory;

(B) creating a virtual device;

(C) whenever a write operation to said mass storage system, said write operation specifying a mass storage write address and data to be written, occurs:

(1) if there is not a block of data associated with said mass storage write address in said preservation memory, placing a copy of said block of data located in said mass storage system at said mass storage write address in said preservation memory; and

(2) writing said data to be written to said mass storage system at the location specified by said mass storage write address;

and

(D) whenever a read operation to said virtual device, said read operation specifying a virtual device read address, occurs:

- (1) if there is not a block of data associated with said virtual device read address in said preservation memory, returning said data of said block of said mass storage system specified by said virtual device read address as the result of said read operation, and
- (2) if there is a block of data associated with said virtual device read address in said preservation memory, returning said block of data from said preservation memory as the result of said read operation.

2. A method as in claim 1, wherein said mass storage system comprises one or more disks.

3. A method as in claim 1, wherein said mass storage system is a partition of a disk.

4. A method as in claim 1, wherein said preservation memory is a random-access memory.

5. A method as in claim 1, wherein said preservation memory is one or more disks.

6. A method as in claim 1, wherein said preservation memory is a partition of a disk.

7. A method as in claim 1, wherein said preservation memory is a file stored on mass storage system.

8. A method as in claim 1, the method further comprising:

(E) whenever a write operation to said virtual device, said write operation specifying a

virtual device write address and data to be written, occurs:

- (1) if there is not a block of data associated with said virtual device write address in said preservation memory, placing in said preservation memory said data to be written, and
- (2) if there is a block of data associated with said virtual device write address in said preservation memory, replacing in said preservation memory that block of data with said data to be written.

9. A method as in claim 1, the method further comprising:

- (F) whenever a read operation to said mass storage system, said read operation specifying a mass storage read address, occurs, returning said data of said block of said mass storage system specified by said mass storage read address as the result of said read operation.

10. A method as in claim 1, said computer configuration further including a block association memory, said block association memory used to associate blocks stored in said preservation memory with said unique addresses.

11. A method as in claim 10, wherein said block association memory contains entries indicating a unique address in said mass storage system and a location in said preservation memory of a block associated with that unique address.

12. A method as in claim 11, wherein blocks of data in said preservation memory are associated with a unique

address by searching said block association memory entries for a matching address.

13. A method as in claim 12, wherein there is not a block of data associated with an address if there is no entry in said block association memory with a matching address.

14. A method as in claim 11, wherein said block association memory contains an entry for each unique address in said mass storage system indicating a location in said preservation memory of a block associated with that unique address.

15. A method as in claim 12, where a special value for said preservation memory location in said entries indicates that there is not a block of data in said preservation memory associated with that address.

16. A method as in claim 1, wherein said digital computer acts as a file server, and said virtual device is exported to other computers.

17. A method as in claim 16, wherein said mass storage system is exported to other computers.

18. A method as in claim 1, said computer configuration including a second mass storage system, the method further comprising:

creating a second virtual device;
whenever a write operation to said second mass storage system, said write operation specifying a second mass storage write address and data to be written, occurs:
(1) if there is not a block of data associated with said second mass storage write address in said preservation memory, placing a copy of said block of data located in said second mass storage system at said

second mass storage write address in
said preservation memory; and

(2) writing said data to be written to said
second mass storage system at the
location specified by said second mass
storage write address;

and

whenever a read operation to said second virtual
device, said read operation specifying a
virtual device read address, occurs:

(1) if there is not a block of data
associated with said virtual device
read address in said preservation
memory, returning said data of said
block of said second mass storage
system specified by said virtual device
read address as the result of said read
operation, and

(2) if there is a block of data associated
with said block read address in said
preservation memory, returning said
block of data from said preservation
memory as the result of said read
operation.

19. A method as in claim 1, said computer
configuration including a second preservation memory, the
method further comprising:

creating a second virtual device;
whenever a write operation to said mass storage
system, said write operation specifying a
mass storage write address and data to be
written, occurs:

(1) if there is not a block of data
associated with said mass storage write

address in said second preservation memory, placing a copy of said block of data located in said mass storage system at said mass storage write address in said second preservation memory; and

- (2) writing said data to be written to said mass storage system at the location specified by said mass storage write address;

and

whenever a read operation to said second virtual device, said read operation specifying a second virtual device read address, occurs:

- (1) if there is not a block of data associated with said second virtual device read address in said second preservation memory, returning said data of said block of said mass storage system specified by said second virtual device read address as the result of said read operation, and
- (2) if there is a block of data associated with said second virtual device read address in said second preservation memory, returning said block of data from said second preservation memory as the result of said read operation.

20. A method for providing a static snapshot of data stored on a mass storage system, the method operating on a computer configuration that includes:

a digital computer;
a mass storage system connected to said digital computer, said mass storage system being

capable of storing blocks of data having unique addresses; and
a preservation memory connected to said digital computer, said preservation memory being capable of storing blocks of data associated with said unique addresses;

the method comprising the following steps:

- (A) clearing said preservation memory so that no copies of blocks of data are in said preservation memory;
- (B) creating a virtual device;
- (C) whenever a write operation to said mass storage system occurs that specifies a mass storage write address and data to be written occurs, performing the following:
 - (1) if there is not a block of data associated with said mass storage write address in said preservation memory, placing a copy of said block of data located in said mass storage system at said mass storage write address in said preservation memory; and
 - (2) writing said data to be written to said mass storage system at the location specified by said mass storage write address;
- and
- (D) whenever a read operation to said virtual device that specifies a virtual device read address occurs, performing the following:
 - (1) if there is not a block of data associated with said virtual device read address in said preservation memory, returning said data of said

block of said mass storage system
specified by said virtual device read
address as the result of said read
operation, and

- (2) if there is a block of data associated
with said virtual device read address
in said preservation memory, returning
said block of data from said
preservation memory as the result of
said read operation.

21. A method for providing a static snapshot of data
stored on a mass storage system, the method operating on a
computer configuration that includes:

a digital computer;
a mass storage system connected to said digital
computer, said mass storage system being
capable of storing blocks of data having
unique addresses; and
a preservation memory connected to said digital
computer, said preservation memory being
capable of storing blocks of data associated
with said unique addresses;

the method comprising the following steps:

- (A) clearing said preservation memory so that no
copies of blocks of data are in said
preservation memory;
- (B) creating a virtual device;
- (C) whenever a write operation to said mass
storage system occurs that specifies a mass
storage write address and data to be written
occurs, performing the following:
 - (1) if there is not a block of data
associated with said mass storage write
address in said preservation memory,

placing a copy of said block of data located in said mass storage system at said mass storage write address in said preservation memory; and

(2) writing said data to be written to said mass storage system at the location specified by said mass storage write address;

(D) whenever a read operation to said virtual device that specifies a virtual device read address occurs, performing the following:

(1) if there is not a block of data associated with said virtual device read address in said preservation memory, returning said data of said block of said mass storage system specified by said virtual device read address as the result of said read operation, and

(2) if there is a block of data associated with said virtual device read address in said preservation memory, returning said block of data from said preservation memory as the result of said read operation;

(E) whenever a write operation to said virtual device occurs that specifies a virtual device write address and data to be written, performing the following:

(1) if there is not a block of data associated with said virtual device write address in said preservation memory, placing in said preservation memory said data to be written, and

(2) if there is a block of data associated with said virtual device write address in said preservation memory, replacing in said preservation memory that block of data with said data to be written; and

(F) whenever a read operation to said mass storage system that specifies a mass storage read address occurs, returning said data of said block of said mass storage system specified by said mass storage read address as the result of said read operation.

22. A system for providing a static snapshot of data stored on a mass storage system on a computer configuration that includes:

a digital computer;
a mass storage system connected to said digital computer, said mass storage system being capable of storing blocks of data having unique addresses; and
a preservation memory connected to said digital computer, said preservation memory being capable of storing blocks of data associated with said unique addresses;

the system comprising:

(A) means for clearing said preservation memory so that no copies of blocks of data are in said preservation memory;
(B) means for creating a virtual device;
(C) whenever a write operation to said mass storage system occurs that specifies a mass storage write address and data to be written occurs, means for performing the following:

- (1) if there is not a block of data associated with said mass storage write address in said preservation memory, placing a copy of said block of data located in said mass storage system at said mass storage write address in said preservation memory; and
- (2) writing said data to be written to said mass storage system at the location specified by said mass storage write address;

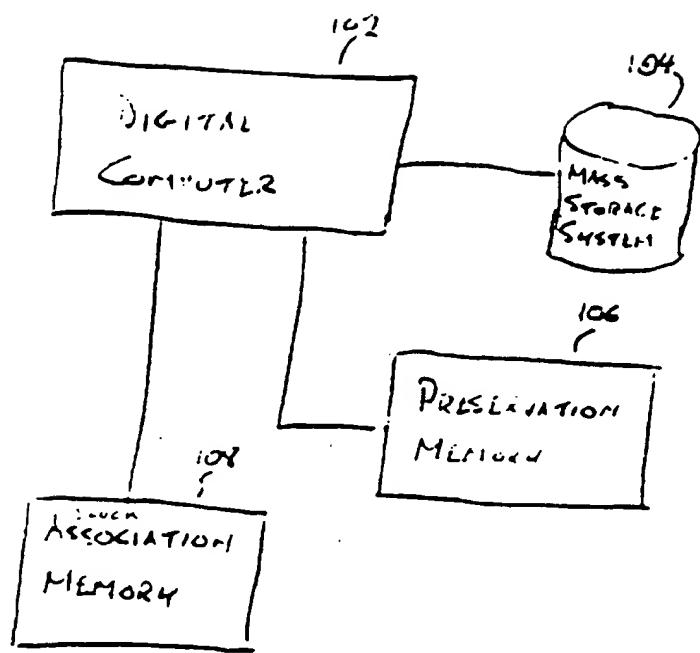
(D) whenever a read operation to said virtual device that specifies a virtual device read address occurs, means for performing the following:

- (1) if there is not a block of data associated with said virtual device read address in said preservation memory, returning said data of said block of said mass storage system specified by said virtual device read address as the result of said read operation, and
- (2) if there is a block of data associated with said virtual device read address in said preservation memory, returning said block of data from said preservation memory as the result of said read operation;

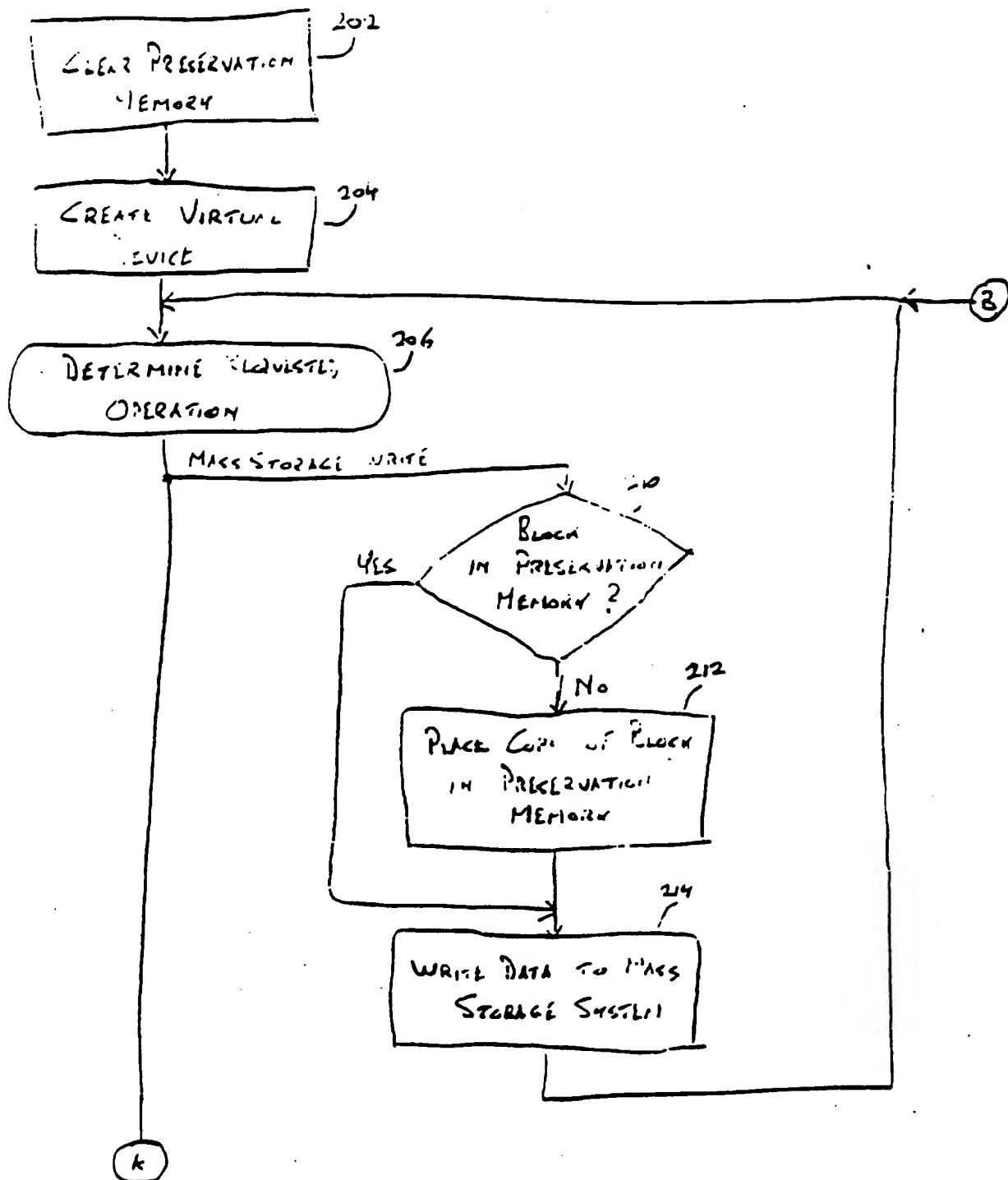
(E) whenever a write operation to said virtual device occurs that specifies a virtual device write address and data to be written, means for performing the following:

- (1) if there is not a block of data associated with said virtual device write address in said preservation memory, placing in said preservation memory said data to be written, and
- (2) if there is a block of data associated with said virtual device write address in said preservation memory, replacing in said preservation memory that block of data with said data to be written; and

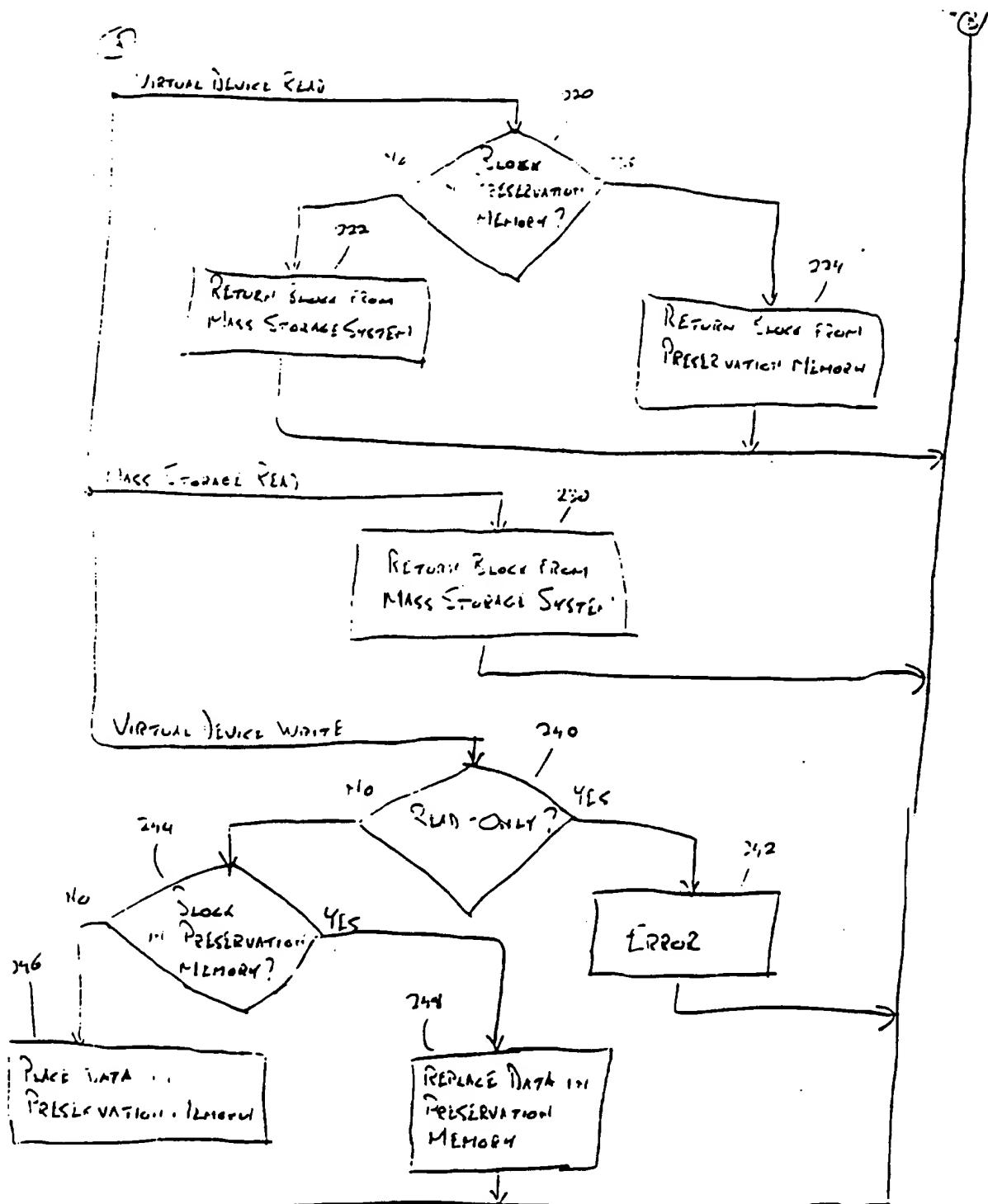
(F) whenever a read operation to said mass storage system that specifies a mass storage read address occurs, means for returning said data of said block of said mass storage system specified by said mass storage read address as the result of said read operation.



- FIGURE 1 -

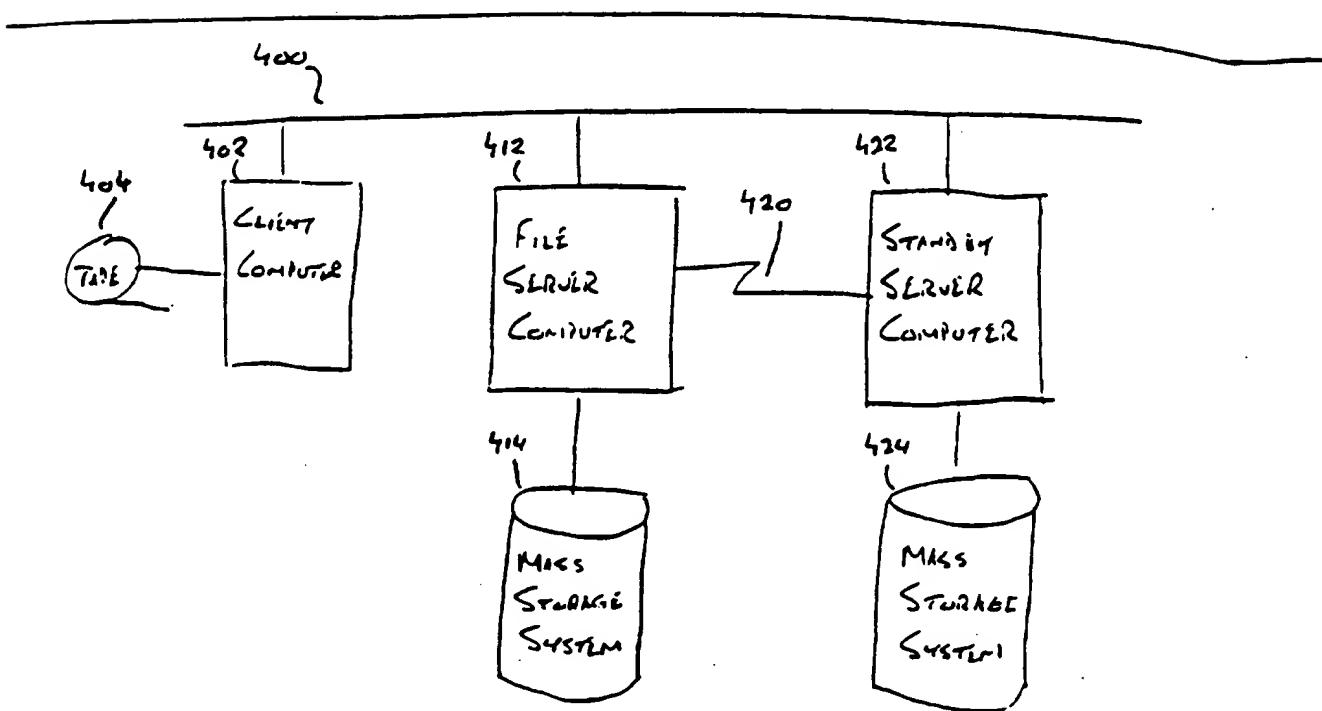
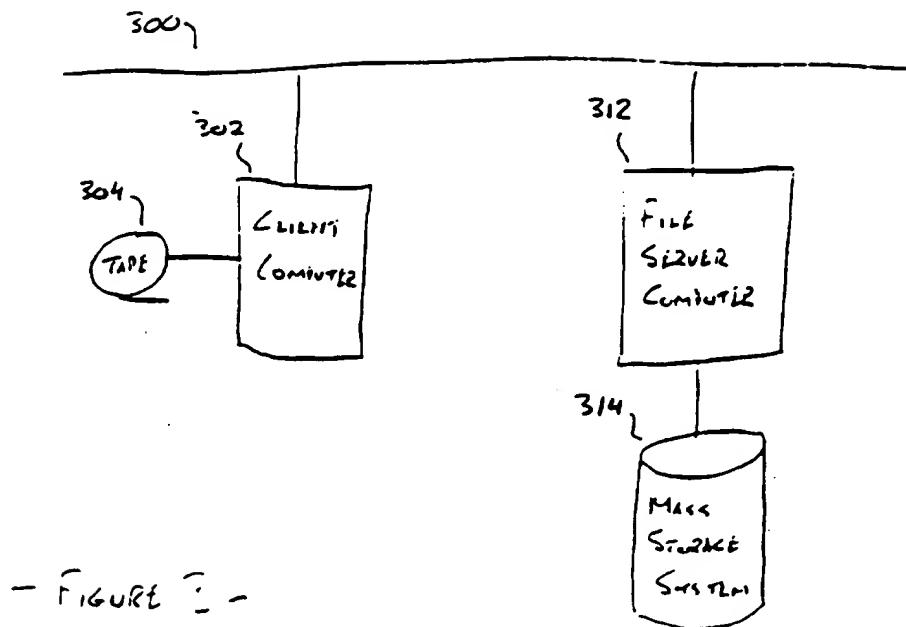


- FIGURE 2 -

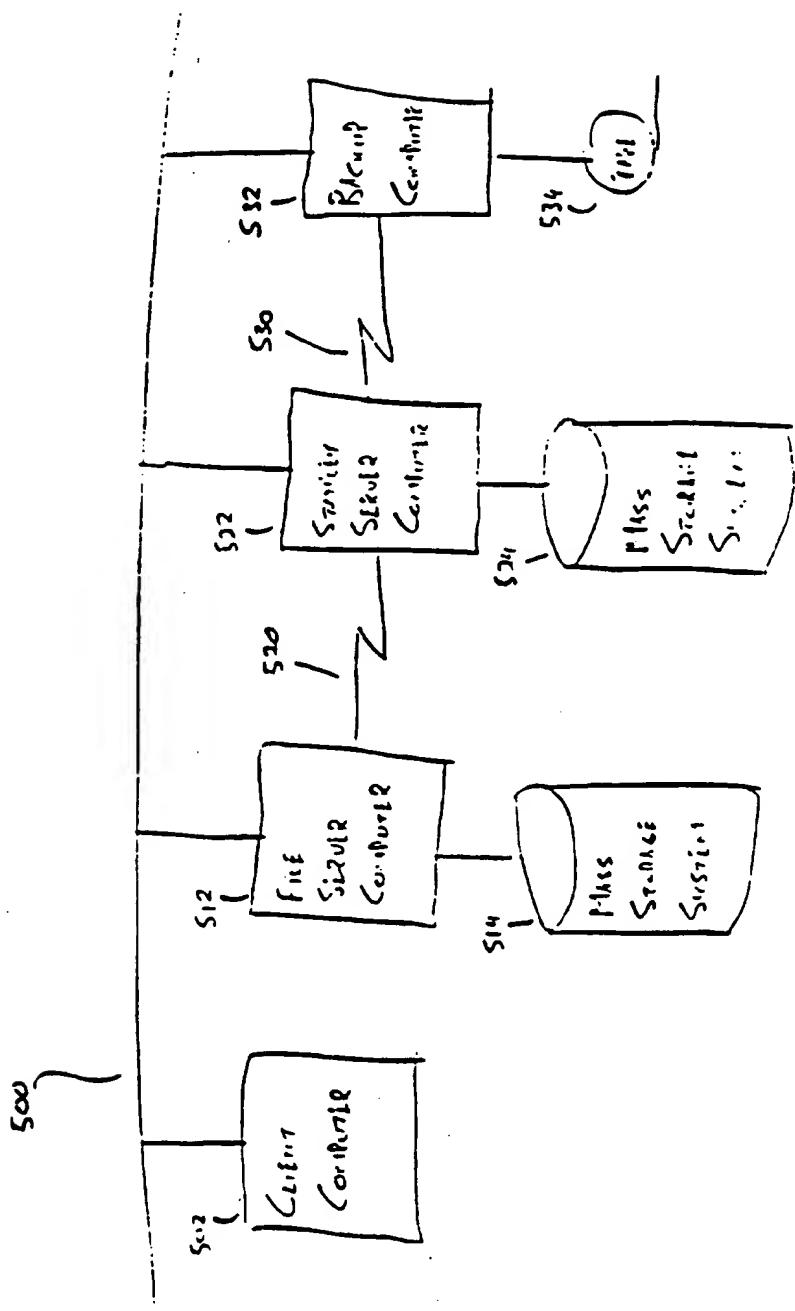


- FIGURE 2 [CONT'D] -

4/5



- FIGURE 4 -



- FIGURE 5 -

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US95/13324

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) : G06F 12/16, 12/08

US CL : 395/468, 488, 462, 412

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 395/468-470, 488-489, 462, 412-413, 417 and 419.

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

Please See Extra Sheet.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X,P	US, A, 5,379,391 (BELSAN ET AL) 03 January 1995, column 1, lines 17-22, 43-49, column 11, line 1 to column 13, line 39. Also, see the abstract.	1-14, 18-22
X,P	US, A, 5,426,747 (WEINREB ET AL) 20 June 1995, column 2, lines 50-60, column 6, line 57 to column 7, line 29, claim 1, column 11, lines 30-38, column 12, lines 10-15, column 14, lines 29-37, column 3, lines 17-21.	1-17, 20-22

Further documents are listed in the continuation of Box C. See patent family annex.

* Special categories of cited documents:	
"A"	document defining the general state of the art which is not considered to be part of particular relevance
"E"	earlier document published on or after the international filing date
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reasons (as specified)
"O"	document referring to an oral disclosure, use, exhibition or other means
"T"	document published prior to the international filing date but later than the priority date claimed
T	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
X	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
Y	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
A	document member of the same patent family

Date of the actual completion of the international search
05 JANUARY 1996

Date of mailing of the international search report
22 MAR 1996

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231
Facsimile No. (703) 305-3230

Authorized officer
ARIO ETIENNE
Telephone No. (703) 308-7562

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US95/13324

B. FIELDS SEARCHED

Electronic data bases consulted (Name of data base and where practicable terms used):

USPTO, Automated Patent Search (APS),
Files: USPAT & JPOAPS

Search terms: snapshot, image, intercept, virtual device, disks, mass storage, cache, flush, clear, files.